

Chapter 7

DOCUMENTING USE CASES

There is a lot of documentation associated with Use Cases that needs to be organized somehow. You want the documentation to be understandable, but you need other things as well. For example, you may want to be able to trace from System level to subordinate Use Cases, or from Use Cases to Scenarios to Test Plans. You may want to be able to set priority levels on a Use Case or any part of a Use Case, then run a report later listing all Use Cases of a certain priority. Or you may want to be able to set and check status on a Use Case, such as has it been coded or tested.

In this Chapter, we'll look at a sample Use Case template, and consider different methods for implementing that template.

Documentation Templates

Below are some sample documentation templates. We start with an overall system description. You will have just one of these documents. It will probably be in a file or document by itself. We follow that with a document template for a Use Case.

System Name

<Brief description. In a large system, this could be several pages. Note this is not meant to be detailed requirements, but a basic overview of the system.>

Risk Factors

<list risk factors here in priority order>

System Level Use Case Diagram

<Or you could just list the Actors and Use Cases in text.>

Architecture Diagram

<Include a description of the interfaces as well. These could be on the diagram, or listed in text.>

Subsystem Descriptions

<Include a brief description of each subsystem.>

Figure 89: System Description Document Template

Next, we need descriptions for the Use Cases. Each Use Case will likely be in a document or file by itself. If a Use Case has a lot of detailed Scenario descriptions, they might also be in separate files. First is a very detailed Use Case document template. You will have one for each system level Use Case. You will also have one for each subordinate Use Case. You will need a similar

document for the detailed Scenarios. Just leave out the parts that are unnecessary or redundant.

Use Case Name

<Brief description. Usually a paragraph or less.>

Actors

<A list of the Actors who communicate with this Use Case>

Priority

<How important is this Use Case to the project?>

Status

<What point are we in developing this Use Case?>

Pre-Conditions

<A list of conditions that must be true before the Use Case starts>

Post-Conditions

<A list of conditions that must be true when the Use Case ends, no matter which Scenario is executed.>

Extension Points

<If the Use Case has extension points, list them here.>

“Used” Use Cases

<If the Use Case uses other Use Cases, list them here.>

Figure 90: Detailed Use Case Description Document Template

Flow of Events

<This could be in text, in an Activity diagram, in a Sequence diagram, or any or all of them.>

Activity Diagram

<An activity diagram of the flow of events, or some significant or complex part of the flow of events.>

User Interface

<For systems which interface with people, include a description of the user interface, possibly using storyboards.>

Scenarios

<They should at least be listed here, but may also include a brief description.>

Sequence Diagrams

<If you don't have separate documents for Scenarios, you might include sequence diagrams for them here.>

Subordinate Use Cases

<If the Use Case has subordinate Use Cases, show them here. Or you could include a Use Case diagram for the subordinate Use Cases. Or both. Also tell what subsystem is responsible for this subordinate Use Case.>

Diagram of Participating Objects

<A class diagram showing all the classes whose objects interact to implement this use case. You can also show interfaces to the Use Case here, and which of the classes implement the interfaces.>

Other Artifacts

<This can include references to the subsystem the Use Case belongs to, an analysis model, a design model, code, or test plans.>

Figure 91: Detailed Use Case Description Document Template (cont.)

You don't have to include all these sections. If your Use Cases are not this complex, then you don't need all these sections. You may find you want additional sections. That's fine. This is just a sample given as a starting point. If it works as is, use it. Otherwise, modify as needed.

Tool Support for Use Case Documents

Unless you are writing Use Cases on paper with pen, pencil, crayon, or whatever, you will be looking for some kind of tool support for your Use Case documents. Consider your needs before buying tools. Use Cases documents could be done in desktop publishing tools such as FrameMaker, word processing tools such as Microsoft Word, spreadsheets such as Microsoft Excel, HTML pages or other web based formats, requirements management tools such as Requisite Pro, groupware such as Lotus Notes, email programs, or stored in a relational database. Almost any tool can be used to generate and store Use Case documents, so you have to consider the needs of your team, project, and company.

Answer these questions before going tool shopping:

- Are you just looking for a way to record text?
- Do you need search capabilities?
- Do you need to share the documents with a small team, a large team, or a geographically distributed team?
- What about report generation? What kinds of reports would you like to get about your Use Cases?
- Do you want to be able to hyperlink from one part of a document to another part or another document?
- Do the documents have to be sharable across multiple platforms?
- Do you need the ability to compose documents into books to get the right page numbering or chapter numbering across documents?
- What other tools might need to use what is in the Use Case documents?
How do you integrate to those tools?

After answering these questions you will be prepared to evaluate the tools you already have, as well as tools you are considering buying. One last thing to consider is ease of use. Hard to use tools tend to sit on a shelf, unless there are no other options. In this case you have many options, so look for something your team will use.

Order Processing System Use Cases

In real life, the Use Cases for a small system will be fairly well worked out before work begins on building the system. Large system tend to work more in parallel than linearly. So some people will work on Use Cases, while others start building code for the Use Cases already completed. Since our Order Processing system is relatively small, we have completed all the Use Cases for it before building the system. But we have not detailed the Scenarios. They seem pretty simple. If we find out later they are relatively complex, then we can write out detailed descriptions.

The rest of this chapter is the Use Case documentation for the order processing system. Don't feel you have to read all of it. Just browse a bit to get a feel for the documentation. Here are a few things to look for in the Use Cases following.

You'll see a lot of uses relationships between Use Cases, but no extends. Uses are found early in the process, and allow you to show commonality between parts of the system. Extends tends to be added later, when you find some new requirement or functionality that extends the current system. Since we haven't built the first system yet, we don't have anything to extend.

We noticed that all of the Use Cases involving the Customer actor required gaining access to the system. So we added a new Use Case called Login. But now we had to figure out it's relationship to the other Use Cases. Our first thought was to have each of the other Use Cases start out by using Login. That feels comfortable if the system is a bunch of independent applications, each with their own interface. So we start the Place Order application, and it calls Login as its first activity. But we don't see this system that way. Instead, there is a common front end for logging into the application. Based on user selection, we branch to one of the other operations. So we have a branch in Login with uses relationships to the other Use Cases.

Look at the activity diagram for Login to see branching shown as a decision point. The activity diagram for Place Order shows a way to diagram an alternative flow of events. This diagram also shows looping.

We're not entirely happy with the process for handling backordered items, but since it seems to work, we'll go with it for now. This is an area we'll plan to revisit later to make the process more efficient. Right now our focus is on getting a working system completed.

Look at what happened to some of the uses relationships. A Use Case must be completely contained within a system, or subsystem, except for communicates

relationships with actors. This means that we cannot have a uses relationship from a Use Case in one subsystem to a Use Case in another subsystem. The Place Order use case in the system diagram uses the Give Product Information use case. But these Use Cases ended up in different subsystems, Place Order in the Manage Orders subsystem, and Give Product Information in the Product Info subsystem. So our uses relationship at the system level became a communicates relationship in the subsystems. You will see this in a number of places in the following pages.

Order Processing System

We are developing order processing software for a mail order company called National Widgets, which is a reseller of products purchased from various suppliers. Twice a year the company publishes a catalog of products, which is mailed to customers and other interested people.

Customers purchase products by submitting a list of products with payment to National Widgets. National Widgets fills the order and ships the products to the customer's address. The order processing software will track the order from the time it is received until product is shipped.

National Widgets will provide quick service. They should be able to ship a customer's order by the fastest, most efficient means possible. Customers may return items for restocking, but will sometimes pay a fee.

Risk Factors

- Some of the people designing the software are inexperienced
- How can we prevent lost orders on system failure?
- The system has to be easy for non-technical people to use.
- Can we be successful if we don't support a web interface?
- What if the system is flooded with orders immediately?
- How do we handle many simultaneous users in different parts of the company?
- How do we handle the database crashing?

System Level Use Cases

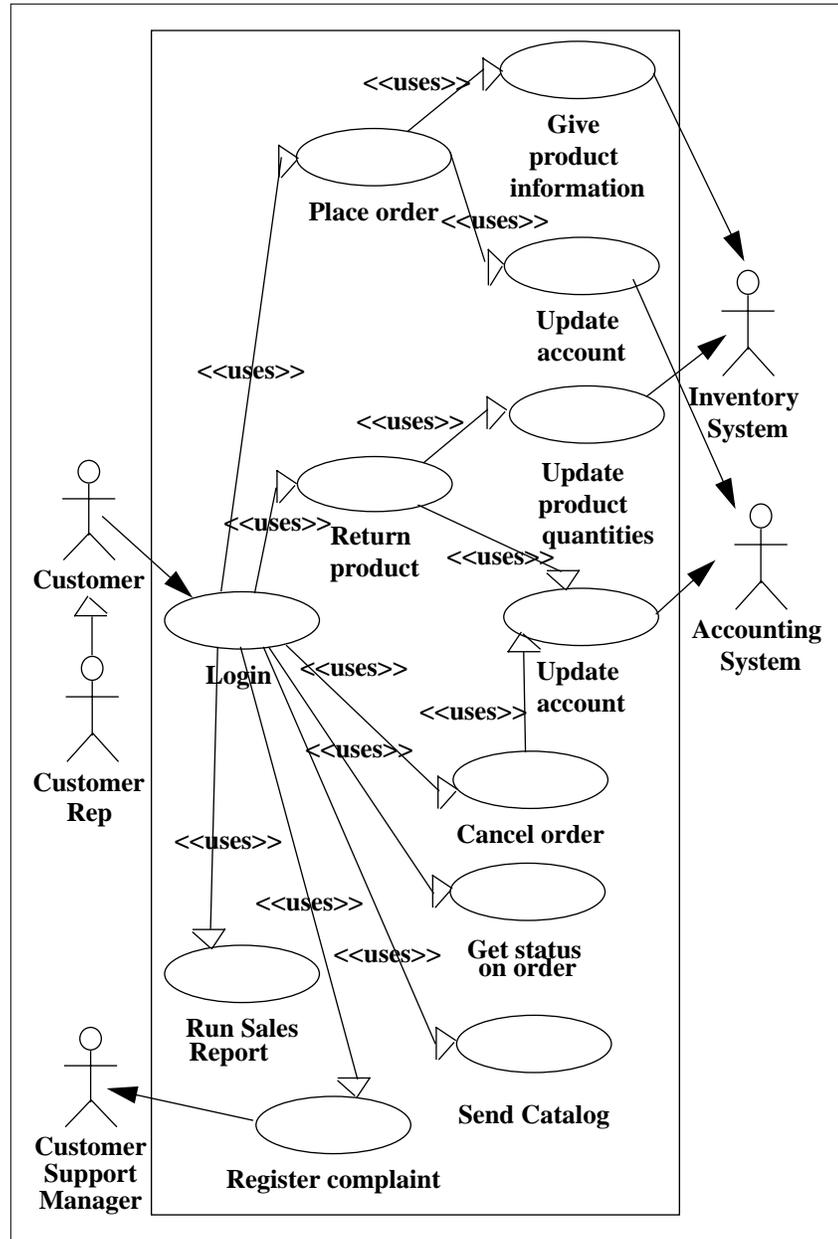


Figure 92: Order Processing Use Cases (1)

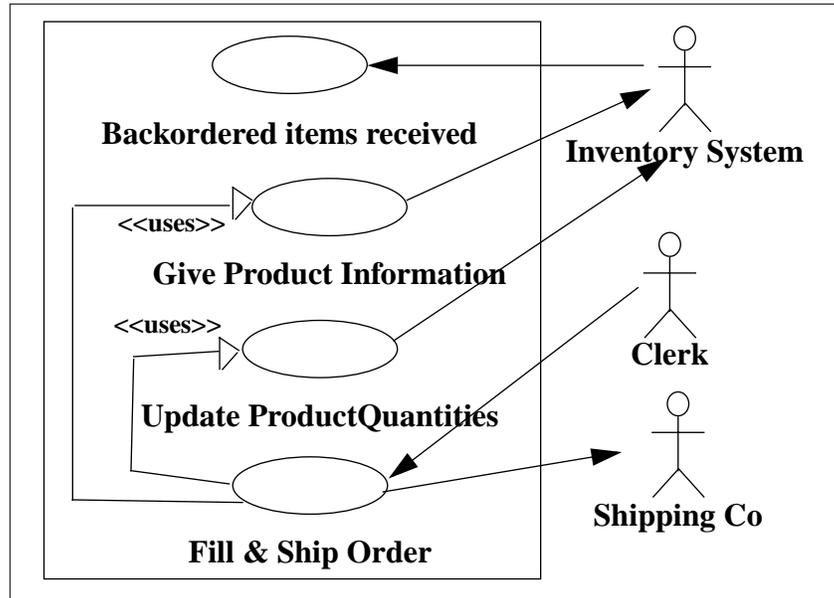


Figure 93: Order Processing Use Cases (2)

Architecture

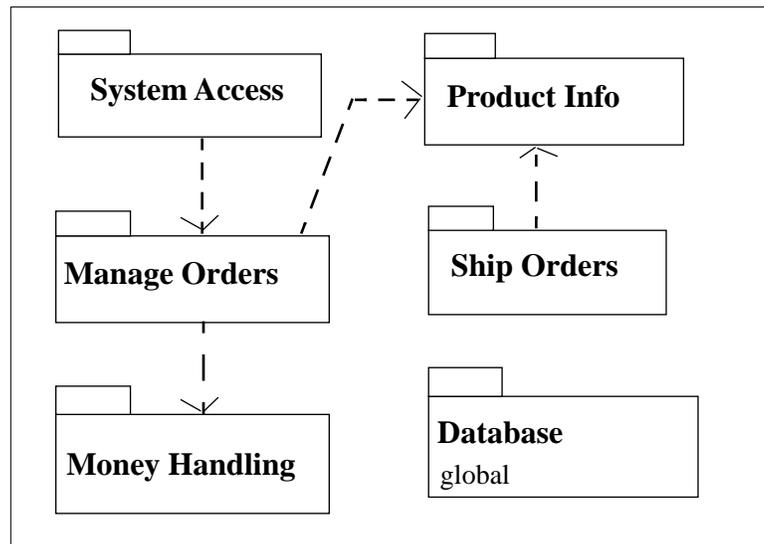


Figure 94: Order Processing Architecture

System Access Subsystem

This subsystem handles log in and log out, and checks for access permissions.

Product Info Subsystem

This subsystem provides all kinds of information about products. It interfaces with the Inventory System.

Manage Orders Subsystem

This subsystem handles order taking, returning of products, status of orders, and canceling orders.

Ship Orders Subsystem

This subsystem prints pick lists for orders, generates mailing labels, and calculates shipping and handling for orders.

Money Handling Subsystem

This subsystem interfaces to the Accounting System, updates accounts, charges and credits customers, and handles checks, credit cards, or money orders.

Database Subsystem

This subsystem contains the data we need to store for the application. It provides standard store, retrieve, update, and delete functions for the data that is stored.

Login

This Use Case describes the process by which users log into the order processing system. It also sets up access permissions for various categories of users.

Actors

- Customer
- Customer Rep

“Used” Use Cases

- Place Order
- Return Product
- Cancel Order
- Get Status on Order
- Send Catalog
- Register Complaint
- Run Sales Report

Flow of Events

1. The Use Case starts when the user starts the application.
2. The system will display the login screen
3. The user enters a username and password.
4. The system will verify the information.
5. The system will set access permissions.
6. The system will display the main screen.
7. The user will select a function
8. While the user does not select Exit loop
9. If the user selects Place Order then
 - 8.1 Use Place Order
- 10.else If the user selects Return Product then
 - 9.1 Use Return Product
- 11.else if the user selects Cancel Order then
 - 10.1 Use Cancel Order
- 12.else if the user selects Review Order then
 - 11.1 Use Get Status on Order
- 13.else if the user selects Send Catalog then
 - 12.1 Use Send Catalog
- 14.else if the user selects Contact Customer Service then
 - 13.1 Use Register Complaint

15.else if the user selects Run Sales Report and user is a Customer Rep then
 14.1 Use Run Sales Report
end if
16.The user will select a function
end loop
17.The Use Case ends

Activity Diagram

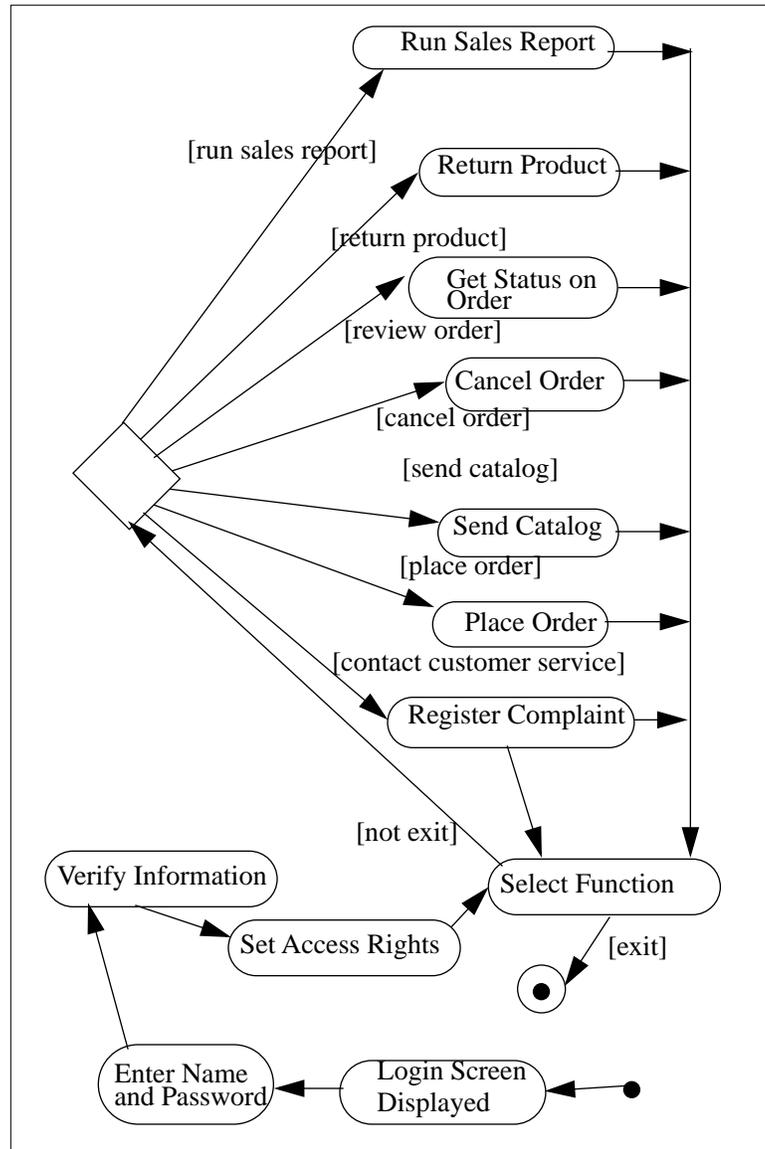
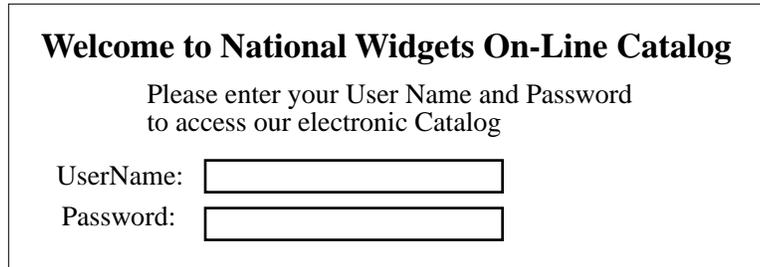


Figure 95: Place Order Primary Scenario

User Interface



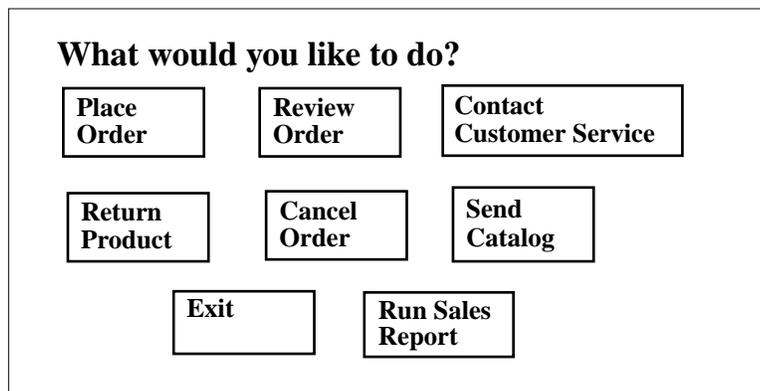
Welcome to National Widgets On-Line Catalog

Please enter your User Name and Password
to access our electronic Catalog

UserName:

Password:

Figure 96: Login Screen



What would you like to do?

Place Order	Review Order	Contact Customer Service
Return Product	Cancel Order	Send Catalog
Exit	Run Sales Report	

Figure 97: Main Screen

Scenarios

- Bad username
- Bad password
- User does not have a username and password for our system
- User selects a function for which they are not allowed access
- User makes no selection of function

Subordinate Use Cases

- System Access - Login
- Manage Orders - Display Main Screen

Place Order

This Use Case describes the process by which orders are entered into the order processing system.

Actors

- Customer
- Customer Rep

“Used” Use Cases

- Give Product Information
- Update Account

Flow of Events

1. The Use Case starts when the customer selects Place Order.
2. The system displays the Place Order Screen
3. The customer enters their name and address.
4. The customer will enter product codes for products they wish to order
5. For each product code entered
 - 4.1 Use Give Product Information
 - 4.2 The system will add the price of the item to the total.
- end
6. The customer will enter credit card payment information.
7. The customer will select submit.
8. The system will verify the information and save the order as pending
9. Use Update Account
10. The order is marked confirmed, an order id is returned to the customer, and the use case ends.

Alternative Flow of Events

At any time before selecting submit, the customer can select cancel. The order is not saved and the use case ends.

Activity Diagram

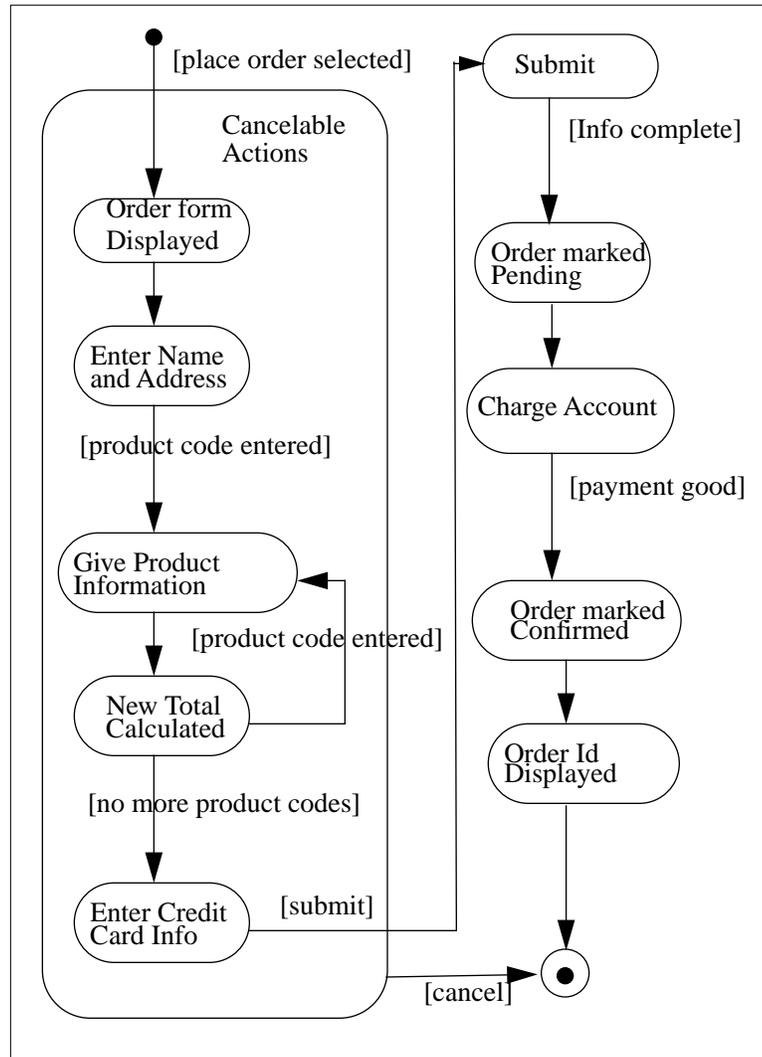


Figure 98: Place Order Primary Scenario

User Interface

Shipping Name	<input type="text"/>			
Shipping Address	<input type="text"/>			
City, State, Zip	<input type="text"/>			
Product Code	Description	Quantity	Price Each	Total Price
			Subtotal	
			Tax	
			Total	
Credit Card Number	<input type="text"/>			
Expiration Date	<input type="text"/>			
Order Id	<input type="text"/>	<input type="button" value="Cancel"/>	<input type="button" value="Submit"/>	

Figure 99: Place Order Screen

Scenarios

- Payment not there
- Shipping address is incomplete
- Product code doesn't match actual products
- Product no longer carried
- Payment bad
- Customer pays by check
- Customer sends order by mail
- Customer phones in order

Subordinate Use Cases

- Manage Orders - Place Order
- Database - Save Order
- Database - Update Order

Give Product Information

This Use Case retrieves product information from the Inventory System.

Actors

- Inventory System

“Used” Use Cases

- None

Flow of Events

1. This Use Case begins when a product code is entered.
2. The system sends a request to the Inventory System for product information based on that product code.
3. The Inventory System returns the product information, what must include at least a product description, price, and stock on hand.
4. The Use Case ends.

Activity Diagram

None.

User Interface

None.

Scenarios

- No such product
- Product no longer carried
- Inventory System unavailable

Subordinate Use Cases

- Product Info - Get Product Info

Update Account

This Use Case interacts with the Accounting System to apply charges or credits to an account.

Actors

- Accounting System

“Used” Use Cases

- None

Flow of Events

1. The Use Case begins when a request to update an account is received.
2. The system will send credit card info and amount of credit or debit to the Accounting System.
3. The Accounting System will send a status of OK.
4. The Use Case ends.

Activity Diagram

None.

User Interface

None.

Scenarios

- Account overdrawn
- Account doesn't exist
- Accounting System not available

Subordinate Use Cases

- Money Handling - Update User Account

Return Product

This Use Case describes the process by which unwanted products are returned to the company by the customer.

Actors

- Customer Rep

“Used” Use Cases

- Search For Order
- Update Product Quantities
- Update Account

Flow of Events

1. The Use Case starts when the customer rep selects return product.
2. Use Search for Order.
3. The system will display the selected order in the Return Products Screen.
4. The customer rep will select the products to return.
5. The customer rep will select submit.
6. Use Update Account.
7. Use Update Product Quantities.
8. The system will display an acknowledgment and the Use Case ends.

Activity Diagram

None.

User Interface

Shipping Name	<input type="text" value="AAA AAAA"/>									
Shipping Address	<input type="text" value="BBB BBBB"/>									
City, State, Zip	<input type="text" value="CCC CCCC"/>									
<table border="1"><thead><tr><th>Product Code</th><th>Description</th><th>Quantity to Return</th></tr></thead><tbody><tr><td>XX</td><td>XXX</td><td></td></tr><tr><td>XX</td><td>XXX</td><td></td></tr></tbody></table>	Product Code	Description	Quantity to Return	XX	XXX		XX	XXX		
Product Code	Description	Quantity to Return								
XX	XXX									
XX	XXX									
<input type="button" value="Submit"/>										

Figure 100: Return Products Screen

Scenarios

- No orders for this customer
- Order id not found in system
- Customer not found in system
- Account no longer valid
- Accounting System not available
- Inventory System not available

Subordinate Use Cases

- Manage Orders - Return Product
- Database - Update Order

Search for Order

This Use Case describes the process for finding a particular order in the system.

Actors

- Customer
- Customer Rep

“Used” Use Cases

None.

Flow of Events

1. The Use Case starts when the Find Order Screen is displayed.
2. The user enters an order id or name and address.
3. The user selects search.
4. If the user entered a name and address
 - 4.1 The system will display a list of orders for that customer, including at least an order id and date of order
 - 4.2 The user will select one order
- end if
5. The system returns an order id and the Use Case ends.

Activity Diagram

None.

User Interface

Find Order

Name

Address

City, State, Zip

Or

Order Id

Figure 101: Find Order Screen

Select An Order

Order Id	Date	Select
XX	XXX	
XX	XXX	

Figure 102: Order Selection Screen

Scenarios

- No such customer
- No orders for customer
- No such order

Subordinate Use Cases

- Manage Orders - Search for Orders
- Database - Get Order List
- Database - Get Order

Update Product Quantities

This Use Case interfaces with the Inventory System to update product quantities in the inventory.

Actors

- Inventory System

“Used” Use Cases

- None

Flow of Events

1. The Use Case starts when a request to Update Product Quantities is received.
2. If the quantity is positive then
 - 2.1 the system will send a request to the Inventory System to add that amount to the stock on hand for a product.else
 - 2.2 the system will send a request to the Inventory System to subtract that amount from the stock on hand for a product.end if
3. The Inventory System will send an acknowledgment and the Use Case ends.

Activity Diagram

None.

User Interface

None.

Scenarios

- Inventory System not available
- Product is no longer stocked

Subordinate Use Cases

- Product Info - Update Product Quantities

Cancel Order

This Use Case describes the process by which a customer can cancel an order.

Actors

- Customer
- Customer Rep

“Used” Use Cases

- Search for Order
- Update Account

Flow of Events

1. The Use Case starts when the customer selects Cancel Order.
2. Use Search for Order.
3. The system will display the cancel order screen for that order.
4. The customer will select Cancel.
5. If the order has not been shipped
 - 5.1 Use Update Accountelse
 - 5.2 Display Return Policies Screenend if
6. The Use Case ends.

Activity Diagram

None.

User Interface

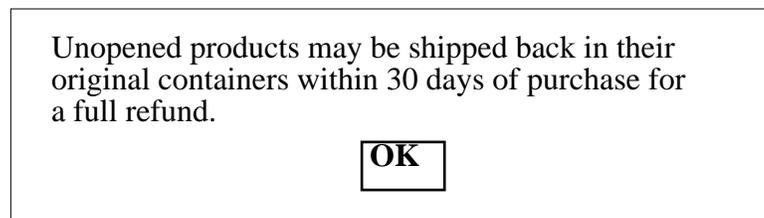


Figure 103: Return Policies Screen

Shipping Name

Shipping Address

City, State, Zip

Product Code	Description	Quantity	Price Each	Total Price

Are you sure you want to cancel this order?

Figure 104: Cancel Order Screen

Scenarios

- No orders for this customer
- Order id not found in system
- Customer not found in system
- Account no longer valid
- Accounting System not available

Subordinate Use Cases

- Manage Orders - Cancel Order
- Database - Update Order

Get Status on Order

This Use Case describes the process by which a customer can get the current status on any of their orders.

Actors

- Customer
- Customer Rep

“Used” Use Cases

- Search for Order

Flow of Events

1. The Use Case starts when the user selects review order.
2. Use Search for Order.
3. The system will display the order status screen and the Use Case ends.

Activity Diagram

None.

User Interface

Shipping Name	<input style="width: 100%;" type="text"/>		
Order Id	<input style="width: 50%;" type="text"/>	Status	<input style="width: 50%;" type="text"/>
		<input type="button" value="OK"/>	
Product	Description	Quantity	Item Status

Figure 105: Get Status on Order Screen

Scenarios

- Order not found

Subordinate Use Cases

- Manage Orders - Get Status on Order

Send Catalog

This Use Case describes how a customer can request a catalog.

Actors

- Customer
- Customer Rep

“Used” Use Cases

- None

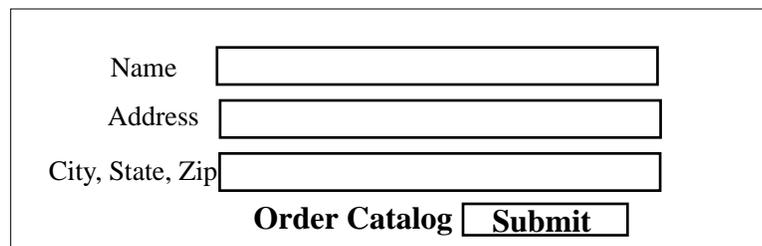
Flow of Events

1. The Use Case starts when the customer selects send catalog.
2. The catalog order screen is displayed.
3. The user enters a name and address
4. The user selects submit
5. The request is submitted as a regular order with no charges to the system.
6. The Use Case ends.

Activity Diagram

None.

User Interface



The user interface for the Order Catalog screen consists of three input fields stacked vertically. The first field is labeled 'Name', the second 'Address', and the third 'City, State, Zip'. Below these fields is a button labeled 'Submit' with the text 'Order Catalog' to its left.

Figure 106: Order Catalog Screen

Scenarios

None.

Subordinate Use Cases

- Manage Orders - Send Catalog
- Database - Save Order

Register Complaint

This Use Case describes how a customer can give feedback to the company.

Actors

- Customer
- Customer Rep
- Customer Support Manager

“Used” Use Cases

- None

Flow of Events

1. The Use Case starts when the user selects contact customer service.
2. The message screen is displayed.
3. The user enters text.
4. The user selects submit.
5. The system sends the text entered in an email message to the customer support manager and the Use Case ends.

Activity Diagram

None.

User Interface

Enter Text of message

Your email address

Figure 107: Message Screen

Scenarios

None.

Subordinate Use Cases

- Manage Orders - Register Complaint

Run Sales Report

This Use Case describes how a customer rep can get reports on sales.

Actors

- Customer Rep

“Used” Use Cases

- None

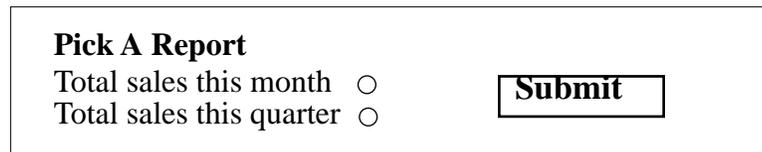
Flow of Events

1. The Use Case starts when the user selects run sales report.
2. The choose report screen is displayed.
3. The user selects a report.
4. The user selects submit.
5. The system displays the report and the Use Case ends.

Activity Diagram

None. The flow is very simple and doesn't need it.

User Interface



Pick A Report
Total sales this month
Total sales this quarter **Submit**

Figure 108: Report Selection Screen

Scenarios

None.

Subordinate Use Cases

- Manage Orders - Run Sales Report
- Database - Get Sales Data

Fill and Ship Order

This Use Case describes how a warehouse clerk gets information on what orders to fill, their products, and the addresses for shipping.

Actors

- Clerk

“Used” Use Cases

- Give Product Info
- Update Product Quantities

Flow of Events

1. The Use Case starts when the clerk starts the fill and ship order application.
2. The system displays the fill order screen with a list of all confirmed orders.
3. While the clerk selects an order
 - 3.1 The system displays the order
 - 3.2 for all the items in the order
 - 3.2.1 User Give Product Info
 - 3.2.2 The system displays stock on hand information for the item
 - 3.2.3 if the item is out of stock
 - 3.2.3.1 The system marks the item backordered.
 - 3.2.3.2 The system sends a backorder request to the inventory system.
 - 3.2.3.3 The order is marked backorders
 - end if
 - end loop
 - 3.3 If there are items which are not backordered and not shipped then
 - 3.3.1 for each such item
 - 3.3.1.1 Use Update Product Quantities
 - 3.3.1.2 Print item and quantity on packing slip
 - 3.3.1.3 mark item shipped
 - end loop
 - 3.3.2 If all items are marked shipped
 - 3.3.2.1 mark the order shipped
 - end if
 - 3.3.2 The system calculates postage due
 - 3.3.3 The system prints a mailing label with the shipping address and postage due
 - 3.3.4 The system sends a notice to the shipping co that packages are ready to be picked up
 - end if
 - end loop

4. The Use Case ends

Activity Diagram

None. The flow is very simple and doesn't need it.

User Interface

Order Id	Date of Order

Figure 109: Orders to Fill Screen

Order Id	<input type="text"/>									
<table border="1"> <thead> <tr> <th>Product</th> <th>Requested</th> <th>Available</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Product	Requested	Available						
Product	Requested	Available								

Figure 110: Order with Stock on Hand

Scenarios

- Inventory System not available
- Printer not available
- Shipping company not available

Subordinate Use Cases

- Ship Orders - Ship and Fill Order
- Database - Get Order List
- Database - Get Order
- Database - Update Order
- Product Info - Backorder Item

Backordered Items Received

This Use Case describes what happens when a shipment of items is received for which the company has back orders.

Actors

- Inventory System

“Used” Use Cases

- None

Flow of Events

1. The use case starts when the Inventory system notifies the system that back-ordered items have been received.
2. The system finds all orders marked backorders, and changes their status to confirmed.
3. For each item marked backordered, clear its state.
4. The Use Case ends.

Activity Diagram

None. The flow is very simple and doesn't need it.

User Interface

None.

Scenarios

None.

Subordinate Use Cases

- Product Info - Backordered Items Received
- Database - Get Order List
- Database - Update Order

System Access Subsystem

This subsystem handles log in and log out, and checks for access permissions.

System Access Use Cases

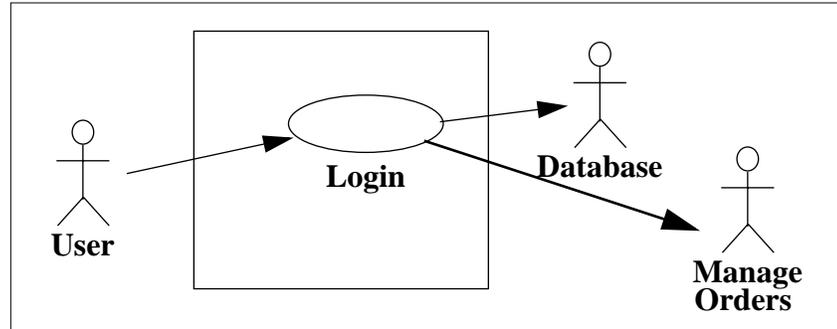


Figure 111: System Access Use Cases

System Access Architecture

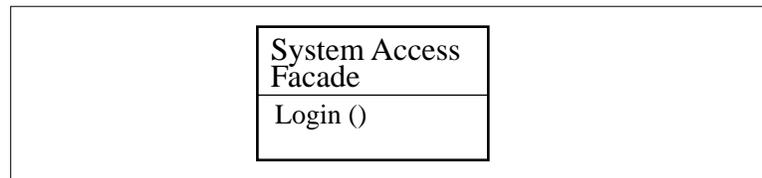


Figure 112: System Access Facade

1. The use case starts when the user starts the application.
2. The system displays the login screen.
3. The user enters a user name and password
4. The system asks the database for a user record for the user name.
5. The system validates the login based on information from the user record.
6. The system sets access levels for the user
7. The system asks Manage Orders to display the main screen.
8. The use case ends

Figure 113: Login Subordinate Use Case

Product Info Subsystem

This subsystem provides all kinds of information about products. It interfaces with the Inventory System.

Product Info Use Cases

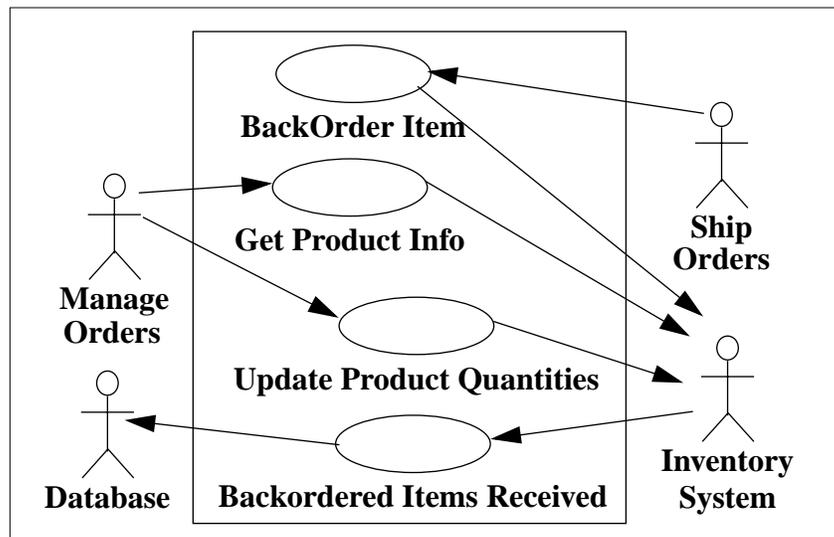


Figure 114: Product Info Use Cases

Product Info Architecture

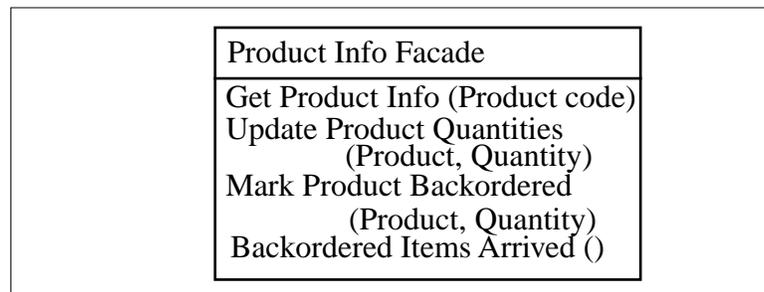


Figure 115: Product Info Facade

1. This Use Case begins when an actor requests product information.
2. The system sends a request to the Inventory System for product information based on that product code.
3. The Inventory System returns the product information, what must include at least a product description, price, and stock on hand.
4. The Use Case ends.

Figure 116: Get Product Info Subordinate Use Case

1. The Use Case starts when a request to Update Product Quantities is received.
2. If the quantity is positive then
 - 2.1 the system will send a request to the Inventory System to add that amount to the stock on hand for a product.
- else
 - 2.2 the system will send a request to the Inventory System to subtract that amount from the stock on hand for a product.
- end if
3. The Inventory System will send an acknowledgment and the Use Case ends.

Figure 117: Update Product Quantities Subordinate Use Case

1. The Use Case starts when a request to backorder items is received
2. The system sends a request to the Inventory System to place an order for out of stock items
3. The Inventory System will send an acknowledgment and the use case ends

Figure 118: BackOrder Item Subordinate Use Case

1. The use case starts when the Inventory system notifies the system that backordered items have been received.
2. The system requests the Database to find all orders marked backorders
3. For each order returned
 - 3.1 The system changes its status to confirmed.
 - 3.2 For each item marked backordered,
 - 3.2.1 The system will clear its state.
 - end loop
 - 3.3 The system requests the Database to update the order.
 - end loop
4. The Use Case ends.

Figure 119: Backordered Items Received Subordinate Use Case

Manage Orders Subsystem

This subsystem handles order taking, returning of products, status of orders, and canceling orders.

Manage Orders Architecture

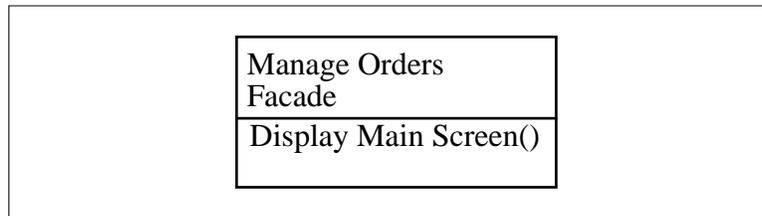


Figure 120: Manage Orders Facade

Manage Orders Use Cases

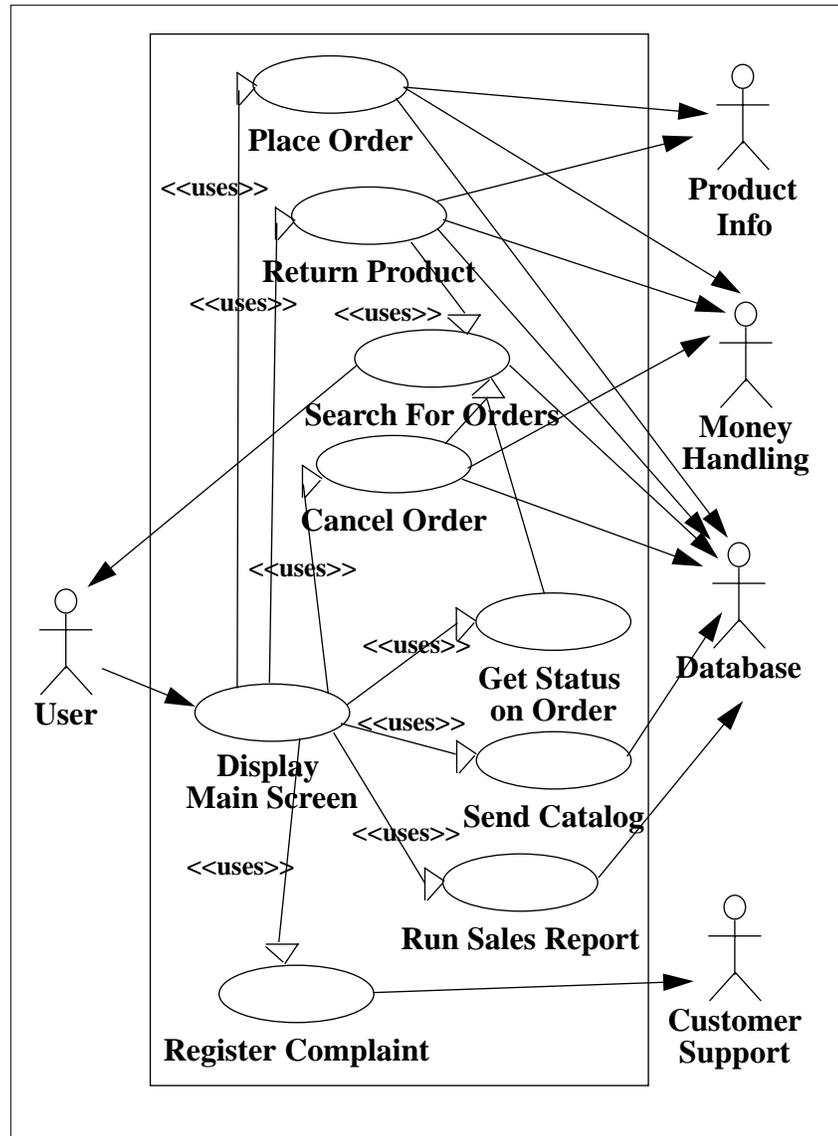


Figure 121: Manage Orders Use Cases

1. The Use Case begins when System Access requests the main screen be displayed.
2. The system will display the main screen.
3. The user will select a function
4. While the user does not select Exit loop
5. If the user selects Place Order then
 8.1 Use Place Order
6. else If the user selects Return Product then
 9.1 Use Return Product
7. else if the user selects Cancel Order then
 10.1 Use Cancel Order
8. else if the user selects Review Order then
 11.1 Use Get Status on Order
9. else if the user selects Send Catalog then
 12.1 Use Send Catalog
- 10.else if the user selects Contact Customer Service then
 13.1 Use Register Complaint
- 11.else if the user selects Run Sales Report and user is a Customer Rep then
 14.1 Use Run Sales Report
- end if
- 12.The user will select a function
- end loop
- 13.The Use Case ends

Figure 122: Display Main Screen Subordinate Use Case

1. The Use Case starts when a Place Order request is received.
2. The system displays the Place Order Screen
3. The customer enters their name and address.
4. The customer enters product codes for products they wish to order
5. For each product code entered
 - 4.1 The system requests product information from Product Info.
 - 4.2 The system adds the price of the item to the total.
- end
6. The customer enters credit card payment information.
7. The customer selects submit.
8. The system verifies the information.
9. The system marks the order pending.
10. The system asks the Database to save the order.
11. The System asks Money Handling to update the user account
12. The system marks the order confirmed
13. The system asks the Database to update the order.
14. The system displays an order id to the customer, and the use case ends.

Alternative Flow of Events

At any time before selecting submit, the customer can select cancel. The order is not saved and the use case ends.

Figure 123: Place Order Subordinate Use Case

1. The Use Case starts when a return product request is received.
2. Use Search for Order.
3. The system will display the selected order in the Return Products Screen.
4. The customer rep will select the products to return.
5. The customer rep will select submit.
6. The system will request Money Handling to Update the user account.
7. The system will request Product Info to Update Product Quantities.
8. The system will request the Database to update the order.
9. The system will display an acknowledgment and the Use Case ends.

Figure 124: Return Product Subordinate Use Case

1. The Use Case starts when a Find Order request is received.
2. The system displays the find order screen.
3. The user enters an order id or name and address.
4. The user selects search.
5. If the user entered a name and address
 - 5.1 The system will ask the database for a list of orders for the customer.
 - 5.2 The system will display a list of orders for that customer, including at least an order id and date of order
 - 5.3 The user will select one orderend if
6. The system asks the Database for an order record
7. The system returns an order id and the Use Case ends.

Figure 125: Search for Orders Subordinate Use Case

1. The Use Case starts when a Cancel Order request is received.
2. Use Search for Order.
3. The system will display the cancel order screen for that order.
4. The customer will select Cancel.
5. If the order has not been shipped
 - 5.1 The system will request Money Handling to Update the user account.
 - 5.2 The system will mark the order cancelled
 - 5.3 The system will request the Database to update the order
- else
6. Display Return Policies Screen
- end if
7. The Use Case ends.

Figure 126: Cancel Orders Subordinate Use Case

1. The Use Case starts when a get status request is received..
2. Use Search for Order.
3. The system will display the order status screen and the Use Case ends.

Figure 127: Get Status on Order Subordinate Use Case

1. The Use Case starts when a send catalog request is received.
2. The catalog order screen is displayed.
3. The user enters a name and address
4. The user selects submit
5. The system creates an order with a product of catalog and no charges
6. The system requests the Database to save the order
7. The Use Case ends.

Figure 128: Send Catalog Subordinate Use Case

1. The Use Case starts when a request to contact customer service is received.
2. The message screen is displayed.
3. The user enters text.
4. The user selects submit.
5. The system sends the text entered in an email message to the customer support manager and the Use Case ends.

Figure 129: Register Complaint Subordinate Use Case

1. The Use Case starts a run sales report request is received.
2. The choose report screen is displayed.
3. The user selects a report.
4. The user selects submit.
5. The system requests sales data from the database.
6. The system displays the report and the Use Case ends.

Figure 130: Run Sales Report Subordinate Use Case

Ship Orders Subsystem

This subsystem prints pick lists for orders, generates mailing labels, and calculates shipping and handling for orders.

Ship Orders Use Cases

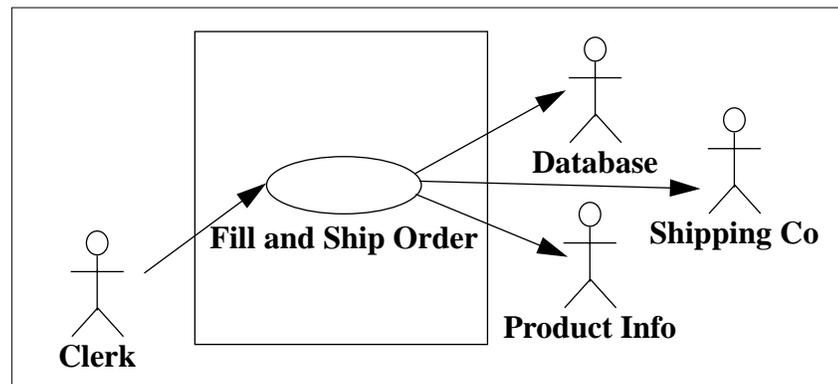


Figure 131: Ship Orders Use Cases

Ship Orders Architecture

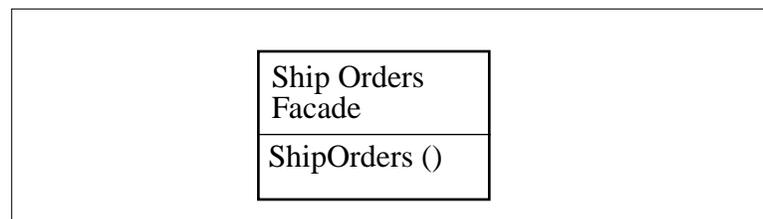


Figure 132: Ship Orders Facade

1. The Use Case starts when the clerk starts the fill and ship order application.
 2. The system displays the fill order screen
 3. The system requests the Database for a list of confirmed orders, by order id.
- Continue on next page

Figure 133: Fill and Ship Order Subordinate Use Case

```
4. While the clerk selects an order
  4.1 The system requests the Database for the order.
  4.2 The system displays the order
  4.3 for all the items in the order
    4.3.1 The system requests stock on hand information from
    Product Info
    4.3.2 The system displays stock on hand information for the
    item
    4.3.3 if the item is out of stock
      4.3.3.1 The system marks the item backordered.
      4.3.3.2 The system requests Product Infor backorder the
      items.
      4.3.3.3 The order is marked backorders
    end if
  end loop

  4.4 If there are items which are not backordered and not shipped
  then
    4.4.1 for each such item
      4.4.1.1 The system requests Product Info to update product
      quantities.
      4.4.1.2 The system will Print item and quantity on packing
      slip
      4.4.1.3 The system will mark the item shipped
    end loop
    4.4.2 If all items are marked shipped
      4.4.2.1 The system will mark the order shipped
    end if
    4.4.3 The system calculates postage due
    4.4.4 The system prints a mailing label with the shipping
    address and postage due
    4.4.5 The system sends a notice to the shipping co that pack-
    ages are ready to be picked up
  end if
  4.5 The system requests the Database to update the order
end loop
5. The Use Case ends
```

Figure 134: Fill and Ship Order Subordinate Use Case (cont)

Money Handling Subsystem

This subsystem interfaces to the Accounting System, updates accounts, charges and credits customers, and handles checks, credit cards, or money orders.

Money Handling Use Cases

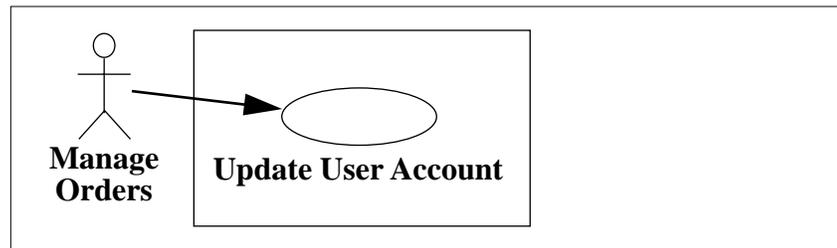


Figure 135: Money Handling Use Cases

Money Handling Architecture

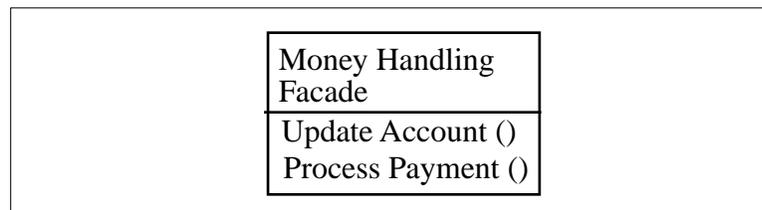


Figure 136: Money Handling Facade

1. The Use Case begins when Manage Orders requests that an account be updated.
2. The system will send credit card info and amount of credit or debit to the Accounting System.
3. The Accounting System will send a status of OK.
4. The Use Case ends.

Figure 137: Update User Account Subordinate Use Case

Database Subsystem

This subsystem contains the data we need to store for the application. It provides standard store, retrieve, update, and delete functions for the data that is stored.

Database Use Cases

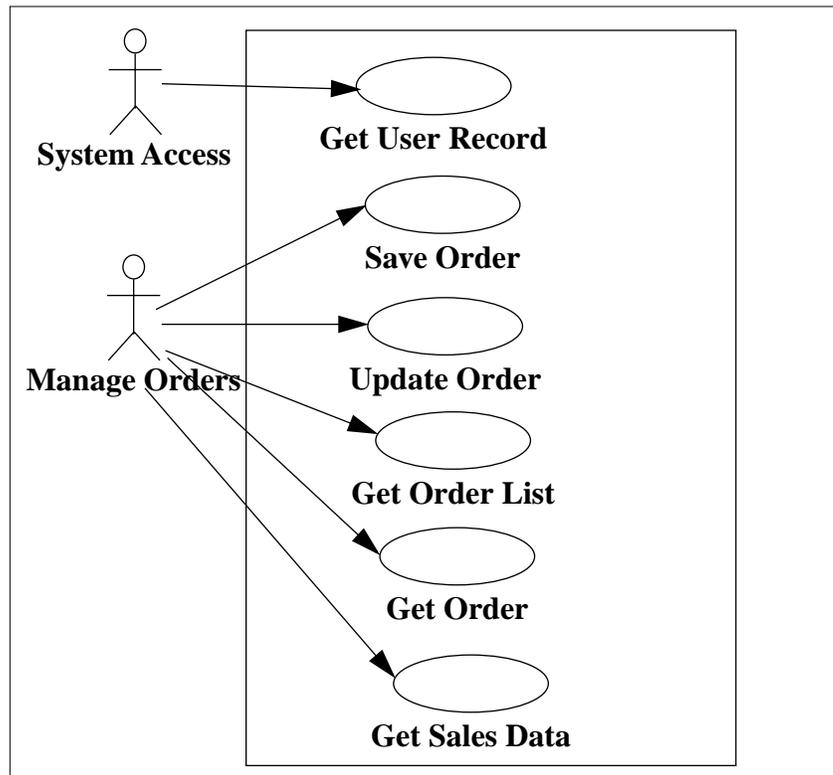


Figure 138: Database Use Cases

All these Use Cases are very similar. We show one as an example

1. The use case starts when System Access requests a User Record.
2. The system locates the user record using the username.
3. The system returns the user record to System Access.
4. The use case ends

Figure 139: Get User Record Subordinate Use Case

Database Architecture

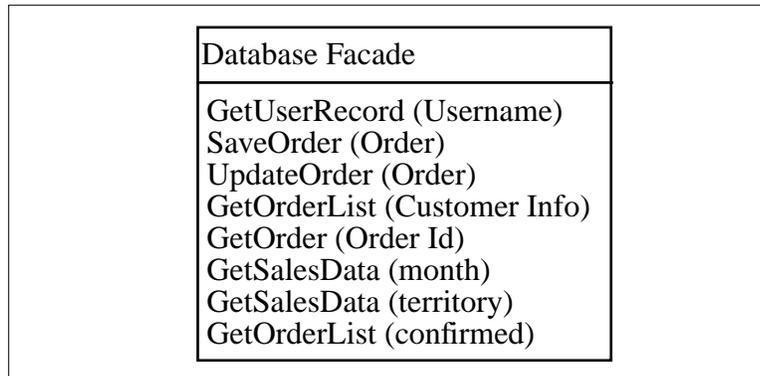


Figure 140: Database Facade